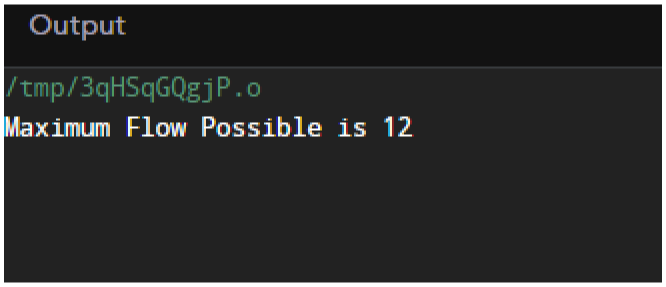
```
#include <iostream>
#include <limits.h>
#include <queue>
#include <string.h>
using namespace std;
#define v 6
int count=0;
bool bfs(int rg1[v][v],int s,int t,int p1[])
{
    bool visited[v];
    int j;
    queue<int> q;
    q.push(s);
    visited[s] = true;
    p1[s] = -1;
    while(!q.empty())
    {
        int i = q.front();
        q.pop();
        for(j = 0;j < v;j++)
        {
        if(visited[j] == false && rg1[i][j] > 0)
        {
            if(j == t)
            {
                p1[j] = i;
                return true;
            }
            q.push(j);
            p1[j] = i;
            visited[j] = true;
        }
    }
}
return false;
}
int fordfulkerson(int graph[v][v],int s,int t)
{
    int i,j,rgraph[v][v],parent[v],maxflow = 0;

    for(int i=0;i<v;i++)
    {
        for(j=0;j<v;j++)
        {
            rgraph[i][j]=graph[i][j];
        }
    }
    int max_flow=0;
    while(bfs(rgraph,s,t,parent)==true)
    {
        count++;
       int path_flow = INT_MAX;
       for(j=t;j!=s;j=parent[j])
       {
           i = parent[j];
           path_flow = min(path_flow,rgraph[i][j]);
       }
       for(j=t;j!=s;j=parent[j])
       {
           i = parent[j];
           rgraph[i][j] -=path_flow;
           rgraph[j][i] += path_flow;
       }
       max_flow+=path_flow;
    }
    return max_flow;
}
int main()
{
    int graph[6][6] =
    {
        {0,16,13,0,0,0},
        {0,0,10,12,0,0},
        {0,4,0,0,14,0},
        {0,0,9,0,0,20},
        {0,0,0,7,0,4},
        {0,0,0,0,0,0}
    };

    cout<<"Maximum Flow Possible is "<<fordfulkerson(graph,0,5);
    return 0;
}
```



```
/tmp/3qHSqGQgjP.o
Maximum Flow Possible is 12
```